BUPT_PRIS at TREC 2014 Knowledge Base Acceleration Track

Yuanyuan Qi, Ye Xu, Dongxu Zhang, Weiran Xu
qiyuanyuan@bupt.edu.cn,bob.ye.xu@gmail.com,
zhangdongxuu@gmail.com ,xuweiran@bupt.edu.cn
Pattern Recognition and Intelligence System, Beijing University of Posts and Telecommunications

Abstract

This paper describes the system in Vital Filtering and Streaming Slot Filling task of TREC 2014 Knowledge Base Acceleration Track. In the Vital Filtering task, The PRIS system focuses attention on query expansion and similarity calculation. The system uses DBpedia as external source data to do query expansion and generates directional documents to calculate similarities with candidate worth citing documents. In the Streaming Slot Filling task, The BUPT_PRIS system utilizes a pattern learning method to do relation extraction and slot filling. Patterns of regular slots which mostly are same to TAC-KBP slots are learned from KBP Slot Filling corpus. Other slots are manually picked up some training seeds for those slot types that KBP didn't contain to use bootstrapping method.

1 Introduction

The goal of KBA track is filtering a large stream of text to find documents that can help update a knowledge base like Wikipedia. The KBA2014 includes three tasks: Vital Filtering(VF) task ,Streaming Slot Filling task and Accelerate & Create. The third task is new open track which is not evaluated. For the Vital Filtering task, given a fixed list of target entities from Wikipedia and Twitter, systems should filter documents worth citing in a profile of the entity. For the Streaming Slot Filling task, given a slot for each of the target entities, systems should detect changes to the slot value, such as location of next performance or founder of an organization. Our team participated in Vital Filtering and Streaming Slot Filling tasks.

2 Vital Filtering

2.1 System Overview

The Vital Filtering (VF) system focuses attention on query expansion, feature generation and vital classification. The framework of our system is shown in Figure 1.

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **NOV 2014**	2. REPORT TYPE	3. DATES COVERED **00-00-2014 to 00-00-2014**

4. TITLE AND SUBTITLE **BUPT_PRIS at TREC 2014 Knowledge Base Acceleration Track**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Beijing University of Posts and Telecommunications,Pattern Recognition and Intelligence System,Beijing 100876,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

presented in the proceedings of the Twenty-Third Text REtrieval Conference (TREC 2014) held in Gaithersburg, Maryland, November 19-21, 2014. The conference was co-sponsored by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA).

14. ABSTRACT

This paper describes the system in Vital Filtering and Streaming Slot Filling task of TREC 2014 Knowledge Base Acceleration Track. In the Vital Filtering task, The PRIS system focuses attention on query expansion and similarity calculation. The system uses DBpedia as external source data to do query expansion and generates directional documents to calculate similarities with candidate worth citing documents. In the Streaming Slot Filling task, The BUPT_PRIS system utilizes a pattern learning method to do relation extraction and slot filling. Patterns of regular slots which mostly are same to TAC-KBP slots are learned from KBP Slot Filling corpus. Other slots are manually picked up some training seeds for those slot types that KBP didn???t contain to use bootstrapping method.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**6**	

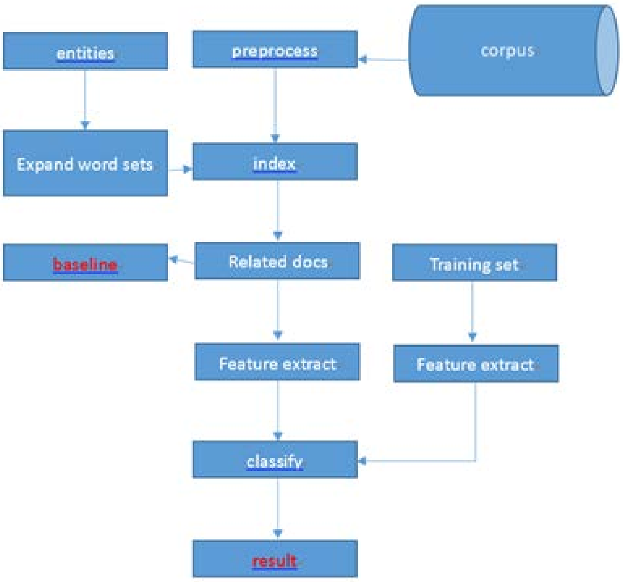

Figure 1 the framework of PRIS system for VF task

First of all, we decompress the corpus and filter the field what we are interested in. Then we indexed the data we have filtered with ElasticSearch. The purpose of indexing is to make it convenient for our algorithms. At the same time we expand the entities with the DBpedia and wiki, then we used the training set to train a classifier,the input is the features we have extracted from the documents.

2.2 Query Expansion

For each entity, the system chooses values of property name and values of property label as expansion terms from the corresponding DBpedia page and label these terms as Name and Label respectively;

If the entity doesn't have a DBpedia page, the system chooses alternative names and the link of homepage as expansion terms from the corresponding twitter page and then visits the homepage to extract key words of homepages as expansion terms too.

2.3 Feature Extraction

To present the document, we extract 10 features of one document as follows: 1.number of target name of an entity; 2.number of redirect name of an entity; 3.number of category of an entity; 4.number of target name in one document; 5.number of redirect name in one document; 6.number of category in one document; 7.An entity's first mention place in the document; 8. An entity's last mention place in the document; 9.length of a document; 10.the cosine similarity of the document and the mean value of related documents of an entity..

The first nine features are calculated directly. The 10th feature is calculated as formula 1[1]

[1] Formula 1 is the document is the mean value of all the related documents in the training set represented by the 10 features.

2.4 Classify

We treat the task as a classify task, so we use three different ways to classify the vital documents: 1. Support Vector Machine (SVM); 2. Random Forest (RF); 3. K-Nearest Neighbor (KNN) and submit all the results generated by the 3 ways. In the Random Forest way, we set the number of trees is 10, and in the KNN way, we make the k=5;

3 Stream Slot Filling

3.1 System overview

For the Streaming Slot Filling task, our system achieved the goal of filling slots by employing a pattern learning and matching method.

This automatic slot filling system contains three steps. First, with query expansion and coreference resolution, we found relative sentences (to make the search faster, we built index using ElasticSearch). Second, we found patterns of slots which are same to TAC-KBP by using KBP training data, and then used bootstrapping method with only single iteration to recall more patterns and also make patterns suit for KBA corpus. Finally, we generated slot answers by matching the patterns and ranking the candidate answers by scoring them with the patterns it was matched. Specially, we manually picked up some training seeds for those slot types that KBP didn’t contain to use bootstrapping method. And there are also some rules found manually such as PER_GENDER, PER_CONTACT, PER_EMAIL, ORG_CONTACT.

The system overview is shown in Fig.2.

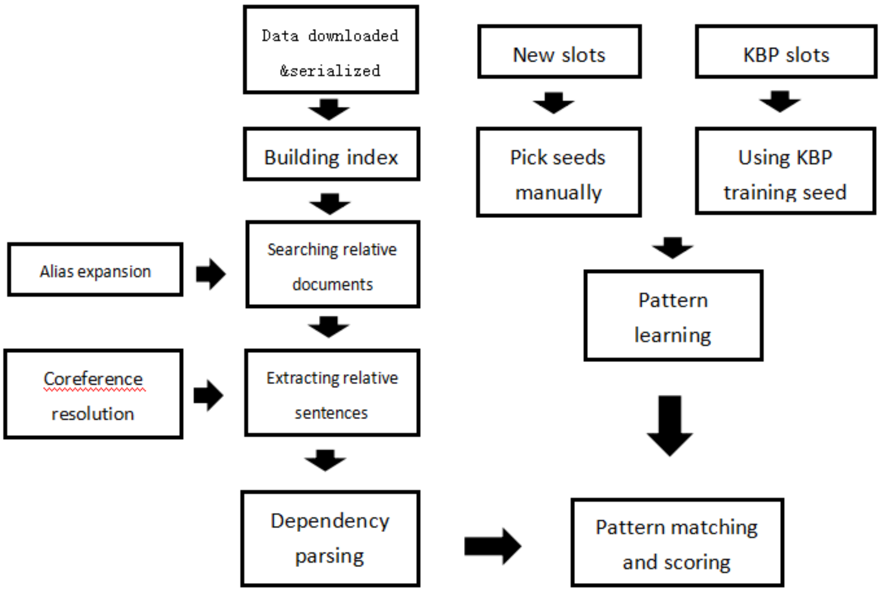

Fig.2. The system overview of SSF task

3.2 Pattern Learning

3.2.1 Find Seed Pattern

We had to find different patterns for those 52 slots separately. Fortunately, we could get training data, formed by sentences with specific query and slot value, for those slots which are in the TAC-KBP slot filling task. For the last several slots, we just collected some training data manually. Then, we used Stanford Parser toolkit to parse each sentence and matched query and slot value on the dependency tree of the sentence. After finding these two nodes, we extracted the path connecting them combined with the entity types of two nodes as our pattern. Notice that we jointed query and slot value which contained more than one word using an underscore so that those words would become only one node on the tree.

3.2.2 Bootstrapping for More Pattern

After gathering pattern seeds, we expanded them on KBA corpus using bootstrapping method so that we could get more patterns to improve the recall and also make our patterns suit for KBA corpus better. Due to unbearable computing time, we only took around 10GB clean text from the official corpus for dependency tree parsing, and implemented bootstrapping method for only one iteration concerning the semantic drift. After gathering lots of patterns by bootstrapping, we pruned them by their frequency of occurrence and literal length.

3.3 Pattern Matching

3.3.1 Find Relative Sentence

For prediction task, firstly we had to find relative sentences mentioned our 109 queries. With trigger words we obtained from task 1 and the coreference resolution information officially supplied, we could search for relative sentences, which we believed would contain most of the answers. Notice that we built an index to speed up our system.

3.3.2 Pattern Matching and scoring

After found those relative sentences and parsed them with Stanford Parser, we could match queries (or alias) and the specific entity type. If both query and slot entity type existed, we would extract the dependency tree path between them and matched that path with our pattern. Then if that path existed in pattern list relative to the entity type, we should add the slot value into our candidate set with the length of the pattern as a weight. After travel through all the relative sentences, we scored those candidates by summing their weights and set a threshold to limit the untrustworthy answers.

4 Result Analysis

Table 1 shows the retrieval performance of our submitted four runs for vital filtering task with useful and vital documents. The primary evaluation metrics for this year’s vital filtering are P(precision),R(recall),F(F-measure) and SU(Scaled Utility).Among all the runs, Run 1 uses the original query without any expansion to search the corpus and submit the retrieval documents. Run 2 uses nonlinear SVM which uses radial basis function as the kernel function. Run 3 uses random forest and with the setting of number of tree is 10. Run 4 uses K-Nearest Neighbor and set the K=5.

We can see from the table that runs using nonlinear SVM have better retrieval performance than others.

Table 1 The performance of submitted runs with useful included

	P	R	F	SU
Run 1	0.837	0.789	0.812	0.808
Run 2	**0.928**	**0.772**	**0.843**	**0.828**
Run 3	0.916	0.723	0.808	0.793
Run 4	0.875	0.240	0.377	0.482

Table 2 shows the retrieval performance of our submitted four runs for vital filtering task with only vital documents.

We can see from the table that runs using random forest have better retrieval performance than others.

Table 2 The performance of submitted runs with vital only

	P	R	F	SU
Run 1	0.185	0.907	0.307	0.000
Run 2	0.201	0.879	0.328	0.000
Run 3	**0.245**	**0.836**	**0.380**	**0.034**
Run 4	0.200	0.245	0.220	0.170

Table 3 shows the retrieval performance of our submitted two runs for Stream Slotting Filling task. The primary evaluation metrics for this year’s Stream Slotting Filling are sokalsneath, cosine, dot and c_TT metrics. The difference between Run 1 and Run 2 is filtering the short patterns.

We can see from the table that Run 2 which has less short pattern gets better retrieval performance than others.

Table 3 the result of SSF with 4 metrics

	sokalsneath metric	cosine metric	dot metric	c_TT metric
Run 1	90.317	41.7237	601.000	380.000
Run 2	**91.517**	**61.1207**	782.000	481.000

5 Conclusion

In this paper, we present our systems for TREC 2014 Knowledge Base Acceleration Track. In the vital filtering task, we apply some traditional classification methods i.e. SVM, random forest , we focus on find ten features to classification to classify big data but we should notice the novel information of documents. In the SSF task, we apply bootstrapping method to find more pattern to find more relative documents.

6 Acknowledgments

The work reported in this paper was supported by 111 Project of China under Grant No. B08004, key project of ministry of science and technology of China under Grant No. 2011ZX03002-005-01, National Natural Science Foundation of China (61273217) and the Ph.D. Programs Foundation of Ministry of Education of China (20130005110004)

References

[1] John R. Frank, Steven J. Bauer,Max Kleiman-Weiner, Daniel A. Roberts, Nilesh Tripuraneni1,Ce Zhang, Christopher Ré,Ellen Voorhees, Ian Soborof. Evaluating Stream Filtering for Entity Profile Updates for TREC 2013

[2]Yan Li, Zhaozhao Wang, Baojin Yu, Yong Zhang, Ruiyang Luo,Weiran Xu, Guang Chen, Jun Guo. PRIS at TREC2012 KBA Track

[3] Chunyun Zhang, Weiran Xu, Ruifang Liu, Weitai Zhang, Dai Zhang, Janshu Ji, Jing Yang. PRIS at TREC2013 Knowledge Base Acceleration Track